Learning to Extract Structured Entities Using Language Models

Haolun Wu[1,2]*, **Ye Yuan**[1,2]*, **Liana Mikaelyan**[3], **Alexander Meulemans**[4], **Xue Liu**[1,2], **James Hensman**[3], **Bhaskar Mitra**[3]

[1] McGill University, [2] Mila - Quebec AI Institute, [3] Microsoft Research, [4] ETH Zürich.

{haolun.wu, ye.yuan3}@mail.mcgill.ca,
xueliu@cs.mcgill.ca, ameulema@ethz.ch,
{t-lmikaelyan, jameshensman, bhaskar.mitra}@microsoft.com.

Abstract

Recent advances in machine learning have significantly impacted the field of information extraction, with Language Models (LMs) playing a pivotal role in extracting structured information from unstructured text. Prior works typically represent information extraction as triplet-centric and use classical metrics such as precision and recall for evaluation. We reformulate the task to be entity-centric, enabling the use of diverse metrics that can provide more insights from various perspectives. We contribute to the field by introducing Structured Entity Extraction and proposing the Approximate Entity Set OverlaP (AESOP) metric, designed to appropriately assess model performance. Later, we introduce a new model that harnesses the power of LMs for enhanced effectiveness and efficiency by decomposing the extraction task into multiple stages. Quantitative and human side-by-side evaluations confirm that our model outperforms baselines, offering promising directions for future advancements in structured entity extraction.

1 Introduction

Information extraction refers to a broad family of challenging natural language processing (NLP) tasks that aim to extract structured information from unstructured text (Cardie, 1997; Eikvil, 1999; Chang et al., 2006; Sarawagi et al., 2008; Grishman, 2015; Niklaus et al., 2018; Nasar et al., 2018; Wang et al., 2018; Martinez-Rodriguez et al., 2020). Examples of information extraction tasks include: *(i)* Named-entity recognition (Li et al., 2020), *(ii)* relation extraction (Kumar, 2017), *(iii)* event extraction (Li et al., 2022), and *(iv)* coreference resolution (Stylianou and Vlahavas, 2021; Liu et al., 2023), as well as higher-order challenges, such as automated knowledge base (KB) and knowledge graph (KG) construction from text (Weikum and Theobald, 2010; Ye et al., 2022; Zhong et al., 2023).

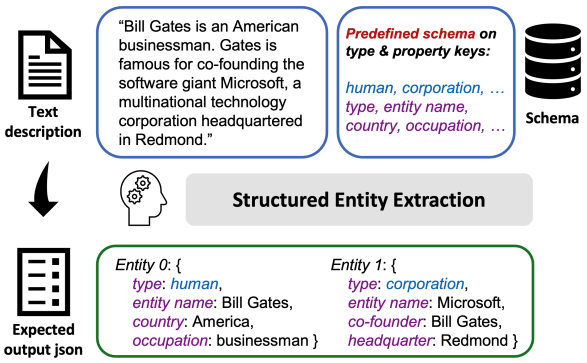

Figure 1: Illustration of the structured entity extraction, an entity-centric formulation of information extraction. Given a text description as well as some predefined schema containing all the candidates of entity types and property keys, we aim to output a structured json for all entities in the text with their information.

The latter may in turn necessitate solving a combination of the former more fundamental extraction tasks as well as require other capabilities like entity linking (Shen et al., 2014, 2021; Oliveira et al., 2021; Sevgili et al., 2022).

Previous formulations and evaluations of information extraction have predominantly centered around the extraction of ⟨*subject, relation, object*⟩ triplets. The conventional metrics used to evaluate triplet-level extraction, such as recall and precision, however, might be insufficient to represent a model's understanding of the text from a holistic perspective. For example, consider a paragraph that mentions ten entities, where one entity is associated with 10 relations as the subject, while each of the other nine entities is associated with only 1 relation as the subject. Imagine a system that accurately predicts all ten triplets for the heavily linked entity but overlooks the other entities. Technically, this system achieves a recall of more than 50% (i.e., 10 out of 19) and a precision of 100%. However, when compared to another system that recognizes one correct triplet for each of the ten entities and achieves the same recall and precision, it becomes

* Equal contribution with random order.

evident that both systems, despite showing identical evaluation scores, offer significantly different insights into the text comprehension. Moreover, implementing entity-level normalization within traditional metrics is not always easy due to challenges like coreference resolution (Stylianou and Vlahavas, 2021; Liu et al., 2023), particularly in scenarios where multiple entities share the same name or lack primary identifiers such as names. Therefore, we advocate for alternatives that can offer insights from diverse perspectives.

In this work, we propose *S*tructured *E*ntity *E*xtraction, an entity-centric formulation of (strict) information extraction, which facilitates diverse evaluations. We define a structured entity as a named entity with associated properties and relationships with other named-entities. Fig. 1 shows an illustration of the structured entity extraction. Given a text description, we aim to first identify the two entities "*Bill Gates*" and "*Microsoft*". Then, given some predefined schema on all possible entity types and property keys (referred to as a *strict* setting in our scenario), the exact types, property keys, property values on all identified entities in the text are expected to be predicted, as well as the relations between these two entities (i.e., *Bill Gates* co-founded *Microsoft*). Such extracted structured entities may be further linked and merged to automatically construct KBs from text corpora. Along with this, we propose a new evaluation metric, *A*pproximate *E*ntity *S*et *O*verla*P* (AESOP), with numerous variants for measuring the similarity between the predicted set of entities and the ground truth set, which is more flexible to include different level of normalization (see default AESOP in Sec. 3 and other variants in Appendix A).

In recent years, deep learning has garnered significant interest in the realm of information extraction tasks. Techniques based on deep learning for entity extraction have consistently outperformed traditional methods that rely on features and kernel functions, showcasing superior capability in feature extraction and overall accuracy (Yang et al., 2022). Building upon these developments, our study employs language models (LMs) to solve structured entity extraction. We introduce a *Mu*lti-stage *S*tructured *E*ntity *E*xtraction (MuSEE) model, a novel architecture that enhances both effectiveness and efficiency. Our model decomposes the entire information extraction task into multiple stages, enabling parallel predictions within each stage for enhanced focus and accuracy. Additionally, we reduce the number of tokens needed for generation, which further improves the efficiency for both training and inference. Human side-by-side evaluations show similar results as our AESOP metric, which not only further confirm our model's effectiveness but also validate the AESOP metric.

In summary, our main contributions are:

- We introduce an entity-centric formulation of the information extraction task within a strict setting, where the schema for all possible entity types and property keys is predefined.
- We propose an evaluation metric, *A*pproximate *E*ntity *S*et *O*verla*P* (AESOP), with more flexibility tailored for assessing structured entity extraction.
- We propose a new model leveraging the capabilities of LMs, improving the effectiveness and efficiency for structured entity extraction.

2 Related work

In this section, we first review the formulation of existing information extraction tasks and the metrics used, followed by a discussion of current methods for solving information extraction tasks.

Information extraction tasks are generally divided into open and closed settings. Open information extraction (OIE), first proposed by Banko et al. (2007), is designed to derive relation triplets from unstructured text by directly utilizing entities and relationships from the sentences themselves, without adherence to a fixed schema. Conversely, closed information extraction (CIE) focuses on extracting factual data from text that fits into a predetermined set of relations or entities, as detailed by Josifoski et al. (2022). While open and closed information extraction vary, both seek to convert unstructured text into structured knowledge, which is typically represented as triplets. These triplets are useful for outlining relationships but offer limited insight at the entity level. It is often assumed that two triplets refer to the same entity if their subjects match. However, this assumption is not always held. Additionally, the evaluation of these tasks relies on precision, recall, and F1 scores at the triplet level. As previously mentioned, evaluating solely on triplet metrics can yield misleading insights regarding the entity understanding. Thus, it is essential to introduce a metric that assesses under-

standing at the entity level through entity-level normalization. In this work, we introduce the AESOP metric, which is elaborated on in Sec. 3.2.

Various strategies have been employed in existing research to address the challenges of information extraction. TextRunner (Yates et al., 2007) initially spearheaded the development of unsupervised methods. Recent progress has been made with the use of manual annotations and Transformer-based models (Vasilkovsky et al., 2022; Kolluru et al., 2020a). Sequence generation approaches, like IMoJIE (Kolluru et al., 2020b) and GEN2OIE (Kolluru et al., 2022), have refined open information extraction by converting it into a sequence-to-sequence task (Cui et al., 2018). GenIE (Josifoski et al., 2022) focuses on integrating named-entity recognition, relation extraction, and entity linking within a closed setting where a knowledge base is provided. Recent work, PIVOINE (Lu et al., 2023), focuses on improving the language model's generality to various (or unseen) instructions for open information extraction, whereas our focus is on designing a new model architecture for improving the effectiveness and efficiency of language model's information extraction in a strict setting.

3 Structured Entity Extraction

In this section, we first describe the structured entity extraction formulation, followed by detailing the Approximate Entity Set OverlaP (AESOP) metric for evaluation. We would like to emphasize that structured entity extraction is not an entirely new task, but rather a novel entity-centric formulation of information extraction.

3.1 Task Formulation

Given a document d, the goal of structured entity extraction is to generate a set of structured entities $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ that are mentioned in the document text. Each structured entity e is a dictionary of property keys $p \in \mathcal{P}$ and property values $v \in \mathcal{V}$, and let $v_{e,p}$ be the value of property p of entity e. In this work we consider only text properties and hence $\mathcal{V}$ is the set of all possible text property values. If a property of an entity is common knowledge but does not appear in the input document, it will not be considered in the structured entity extraction. Depending on the particular situation, the property values could be other entities, although this is not always the case.

So, the goal then becomes to learn a function $f : d \rightarrow \mathcal{E}' = \{e'_1, e'_2, \ldots, e'_m\}$, and we expect the predicted set $\mathcal{E}'$ to be as close as possible to the target set $\mathcal{E}$, where the closeness is measured by some similarity metric $\Psi(\mathcal{E}', \mathcal{E})$. Note that the predicted set of entities $\mathcal{E}'$ and the ground-truth set $\mathcal{E}$ may differ in their cardinality, and our definition of Ψ should allow for the case when $|\mathcal{E}'| \neq |\mathcal{E}|$. Finally, both $\mathcal{E}'$ and $\mathcal{E}$ are unordered sets and hence we also want to define Ψ to be order-invariant over $\mathcal{E}'$ and $\mathcal{E}$. As we do not need to constrain f to produce the entities in any strict order, it is reasonable for Ψ to assume the most optimistic assignment of $\mathcal{E}'$ with respect to $\mathcal{E}$. We denote $\vec{E}'$ and $\vec{E}$ as some arbitrary but fixed ordering over items in prediction set $\mathcal{E}'$ and ground-truth set $\mathcal{E}$ for allowing indexing.

3.2 Approximate Entity Set OverlaP (AESOP) Metric

We propose a formal definition of the Approximate Entity Set OverlaP (AESOP) metric, which focuses on the entity-level and more flexible to include different level of normalization:

$$\Psi(\mathcal{E}', \mathcal{E}) = \frac{1}{\mu} \bigoplus_{i,j}^{m,n} \mathbf{F}_{i,j} \cdot \psi_{\text{ent}}(\vec{E}'_i, \vec{E}_j), \tag{1}$$

which is composed of two phases: *(i) optimal entity assignment* for obtaining the assignment matrix $\mathbf{F}$ to let us know which entity in $\mathcal{E}'$ is matched with which one in $\mathcal{E}$, and *(ii) pairwise entity comparison* through $\psi_{\text{ent}}(\vec{E}'_i, \vec{E}_j)$, which is a similarity measure defined between any two arbitrary entities e' and e. We demonstrate the details of these two phases in this section. We implement Ψ as a linear sum $\bigoplus$ over individual pairwise entity comparisons ψ_{ent}, and μ is the maximum of the sizes of the target set and the predicted set, i.e., $\mu = \max\{m, n\}$.

Phase 1: Optimal Entity Assignment. The optimal entity assignment is directly derived from a matrix $\mathbf{F} \in \mathbb{R}^{m \times n}$, which is obtained by solving an assignment problem between $\mathcal{E}'$ and $\mathcal{E}$. Here, the matrix $\mathbf{F}$ is a binary matrix where each element $\mathbf{F}_{i,j}$ is 1 if the entity $\vec{E}'_i$ is matched with the entity $\vec{E}_j$, and 0 otherwise. Before formulating the assignment problem, we first define a similarity matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$ where each element $\mathbf{S}_{i,j}$ quantifies the similarity between the i-th entity in $\vec{E}'$ and the j-th entity in $\vec{E}$ for the assignment phase. For practical implementation, we ensure inclusion of the union set of property keys from both the i-th entity in $\vec{E}'$ and the j-th entity in $\vec{E}$ for each of these entities. When a property key is absent,

its corresponding property value is set to be an empty string. The similarity is then computed as a weighted average of the Jaccard index (Murphy, 1996) for the list of tokens of the property values associated the same property key in both entities. The Jaccard index involved empty strings is defined as zero in our case. We assign a weight of 0.9 to the entity name, while all other properties collectively receive a total weight of 0.1. This ensures that the entity name holds the highest importance for matching, while still acknowledging the contributions of other properties. Then the optimal assignment matrix $\mathbf{F}$ is found by maximizing the following equation:

$$\mathbf{F} = \arg\max_{\mathbf{F}} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{F}_{i,j} \cdot \mathbf{S}_{i,j}, \tag{2}$$

subject to the following four constraints to ensure one-to-one assignment between entities in the prediction set and the ground truth set: *(i)* $\mathbf{F}_{i,j} \in \{0,1\}$; *(ii)* $\sum_{i=1}^{m} \mathbf{F}_{i,j} \leq 1, \forall j \in \{1,2,\ldots,n\}$; *(iii)* $\sum_{j=1}^{n} \mathbf{F}_{i,j} \leq 1, \forall i \in \{1,2,\ldots,m\}$; *(iv)* $\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{F}_{i,j} = \min\{m,n\}$.

Phase 2: Pairwise Entity Comparison. After obtaining the optimal entity assignment, we focus on the pairwise entity comparison. We define $\psi_{\text{ent}}(\vec{E'}_i, \vec{E}_j)$ as a similarity metric between any two arbitrary entities e' and e from $\mathcal{E}'$ and $\mathcal{E}$.

The pairwise entity similarity function ψ_{ent} is defined as a linear average $\bigotimes$ over individual pairwise property similarity ψ_{prop} as follows:

$$\psi_{\text{ent}}(e', e) = \bigotimes_{p \in \mathcal{P}} \psi_{\text{prop}}(v_{e',p}, v_{e,p}), \tag{3}$$

where $\psi_{\text{prop}}(v_{e',p}, v_{e,p})$ is defined as the Jaccard index between the lists of tokens of the predicted values and ground-truth values for corresponding properties. We define the score as zero for missing properties.

It should be noted that while both $\mathbf{S}$ and ψ_{ent} are used to calculate similarities between pairs of entities, they are not identical. During the entity assignment phase, it is more important to make sure the entity names are aligned, while it is more acceptable to treat all properties equally without differentiation during the pairwise entity comparison. The separation in the definitions of two similarity measures allows us to tailor our metric more precisely to the specific requirements of each phase of the process. Different variants for our proposed AESOP metric are elaborated in Appendix A. We discuss the relationship between traditional metrics, such as precision and recall, and AESOP in Appendix B.

4 Multi-stage Structured Entity Extraction using Language Models

In this section, we elaborate on the methodology for structured entity extraction using LMs. We introduce a novel model architecture leveraging LMs, ***MuSEE***, for ***Mu***lti-stage ***S***tructured ***E***ntity ***E***xtaction. MuSEE is built on an encoder-decoder architecture, whose pipeline incorporates two pivotal enhancements to improve effectiveness and efficiency: *(i) reducing output tokens* through introducing additional special tokens where each can be used to replace multiple tokens, and *(ii) multi-stage parallel generation* for making the model focus on a sub-task at each stage where all predictions within a stage can be processed parallelly.

Reducing output tokens. Our model condenses the output by translating entity types and property keys into unique, predefined tokens. Specifically, for the entity type, we add prefix "**ent_type_**", while for each property key, we add prefix "**pk_**". By doing so, the type and each property key on an entity is represented by a single token, which significantly reduces the number of output tokens during generation thus improving efficiency. For instance, if the original entity type is "*artificial object*" which is decomposed into 4 tokens (i.e., "*_art*", "*if*", "*ical*", "*_object*") using the T5 tokenizer, now we only need one special token, "**ent_type**_*artifical_object*", to represent the entire sequence. All of these special tokens can be derived through the knowledge of some predefined schema before the model training.

Multi-stage parallel generation. In addition to reducing the number of generated tokens, MuSEE further decomposes the generation process into three stages: *(i)* identifying all entities, *(ii)* determining entity types and property keys, and *(iii)* predicting property values. To demonstrate this pipeline more clearly, we use the same text shown in Fig. 1 as an example to show the process of structured entity extraction as follows:

Stage 1: Entity Identification.

❖ [Text Description] ⇒ MuSEE ⇒ *pred_ent_names*
"Bill Gates" "Microsoft" ⟨**EOS**⟩

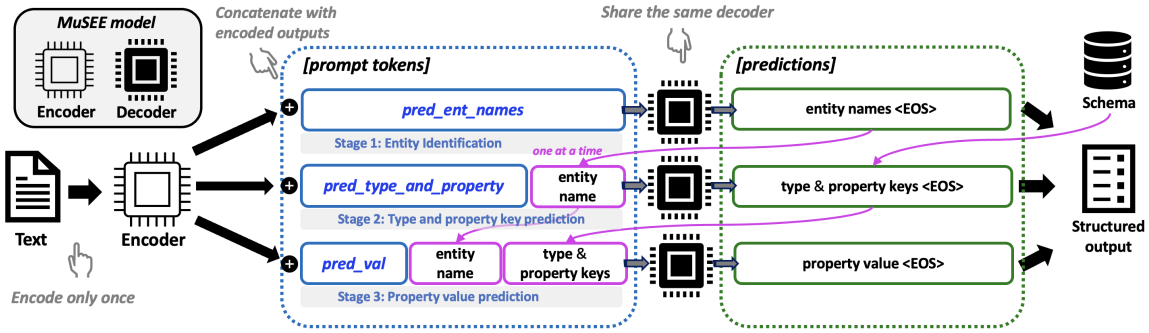

Figure 2: The pipeline of our proposed MuSEE model, which is built on an encoder-decoder architecture. The input text only needs to be encoded once. The decoder is shared for all the three stages. All predictions within each stage can be processed in batch, and teacher forcing enables parallelization even across stages during training.

Stage 2: Type and property key prediction.

- [Text Description] ⇒ MuSEE ⇒ *pred_type_and_property* {"Bill Gates"} **ent_type_human pk_country pk_occupation ⟨EOS⟩**
- [Text Description] ⇒ MuSEE ⇒ *pred_type_and_property* {"Microsoft"} **ent_type_corporation pk_cofounder pk_headquarter ⟨EOS⟩**

Stage 3: Property value prediction.

- [Text Description] ⇒ MuSEE ⇒ *pred_val* {"Bill Gates"} {ent_type_human} {pk_country} **America ⟨EOS⟩**
- [Text Description] ⇒ MuSEE ⇒ *pred_val* {"Bill Gates"} {ent_type_human} {pk_occupation} **Businessman ⟨EOS⟩**
- [Text Description] ⇒ MuSEE ⇒ *pred_val* {"Microsoft"} {ent_type_corporation} {pk_cofounder} **Bill Gates ⟨EOS⟩**
- [Text Description] ⇒ MuSEE ⇒ *pred_val* {"Microsoft"} {ent_type_corporation} {pk_headquarter} **Redmond ⟨EOS⟩**

Among the three stages depicted, *pred_ent_names*, *pred_type_and_property*, and *pred_val* are special tokens to indicate the task. For each model prediction behavior, the first "⇒" indicates inputting the text into the encoder of MuSEE, while the second "⇒" means inputting the encoded outputs into the decoder. All tokens in blue are the prompt tokens input into the decoder which do not need to be predicted, while all tokens in **bold** are the model predictions. For the stage 1, we emphasize that MuSEE outputs a unique identifier for each entity in the given text. Taking the example in Fig. 1, the first stage outputs "*Bill Gates*" only, rather than both "*Bill Gates*" and "*Gates*". This requires the model implicitly learn how to do coreference resolution, namely learning that "*Bill Gates*" and "*Gates*" are referring to the same entity. Therefore, our approach uses neither surface forms, as the outputs of the first stage are unique identifiers, nor the entity titles followed by entity linkings. Notice that we do not need to predict the value for "*type*" and "*name*" in stage 3, since the type can be directly derived from the "**ent_type_**" special key itself, and the name is obtained during stage 1. The tokens in the bracket "{..}" are also part of the prompt tokens and are obtained in different ways during training and inference. During training, these inputs are obtained from the ground truth due to the teacher forcing technique (Raffel et al., 2023). During inference, they are obtained from the output predictions from the previous stages. The full training loss is a sum of three cross-entropy losses, one for each stage. An illustration of our model's pipeline is shown in Fig. 2.

Benefits for Training and Inference. MuSEE's unique design benefits both training and inference. In particular, each stage in MuSEE is finely tuned to concentrate on a specific facet of the extraction process, thereby enhancing the overall effectiveness. Most importantly, all predictions within the same stage can be processed in batch thus largely improving efficiency. The adoption of a teacher forcing strategy enables parallel training even across different stages, further enhancing training efficiency. During inference, the model's approach to breaking down long sequences into shorter segments significantly reduces the generation time. It is also worthy to mention that each text in the above three stages needs to be encoded only once by the MuSEE's encoder, where the encoded output is repeatedly utilized across different stages. This streamlined approach ensures a concise and clear delineation of entity information, facilitating the transformation of unstructured text into a manageable and structured format.

5 Experiments

In this section, we describe the datasets used in our experiment, followed by the discussion of baseline methods and training details.

5.1 Data

In adapting the structured entity extraction, we repurpose the NYT (Riedel et al., 2010), CoNLL04 (Roth and Yih, 2004), and REBEL (Huguet Cabot and Navigli, 2021) datasets, which are originally developed for relation extractions. For NYT and CoNLL04, since each entity in these two datasets has a predefined type, we simply reformat them to our entity-centric formulation by treating the subjects as entities, relations as property keys, and objects as property values. REBEL connects entities identified in Wikipedia abstracts as hyperlinks, along with dates and values, to entities in Wikidata and extracts the relations among them. For entities without types in the REBEL dataset, we categorize their types as "*unknown*". Additionally, we introduce a new dataset, named Wikidata-based. The Wikidata-based dataset is crafted using an approach similar to REBEL but with two primary distinctions: *(i)* property values are not necessarily entities; *(ii)* we simplify the entity types by consolidating them into broader categories based on the Wikidata taxonomy graph, resulting in less specific types. The processes for developing the Wikidata-based dataset is detailed in Appendix C. Comprehensive statistics for all four datasets are available in Appendix D.

5.2 Baseline

We benchmark our methodology against two distinct classes of baseline approaches. The first category considers adaptations from general seq2seq task models: *(i)* LM-JSON: this approach involves fine-tuning pre-trained language models. The input is a textual description, and the output is the string format JSON containing all entities. The second category includes techniques designed for different information extraction tasks, which we adapt to address our challenge: *(ii)* GEN2OIE (Kolluru et al., 2022), which employs a two-stage generative model initially outputs relations for each sentence, followed by all extractions in the subsequent stage; *(iii)* IMoJIE (Kolluru et al., 2020b), an extension of CopyAttention (Cui et al., 2018), which sequentially generates new extractions based on previously extracted tuples; *(iv)* GenIE (Josifoski et al., 2022), an end-to-end autoregressive generative model using a bi-level constrained generation strategy to produce triplets that align with a predefined schema for relations. GenIE is crafted for the closed information extraction, so it includes a entity linking step. However, in our strict setting, there is only a schema of entity types and relations. Therefore, we repurpose GenIE for our setting by maintaining the constrained generation strategy and omitting the entity linking step.

5.3 Training

We follow existing studies (Huguet Cabot and Navigli, 2021) to use the encoder-decoder architecture in our experiment. We choose the T5 (Raffel et al., 2023) series of LMs and employ the pre-trained T5-Base (T5-B) and T5-Large (T5-L) as the base models underlying every method discussed in section 5.2 and our proposed MuSEE. LM-JSON and MuSEE are trained with the Low-Rank Adaptation (Hu et al., 2021), where $r = 16$ and $\alpha = 32$. For GEN2OIE, IMoJIE, and GenIE, we follow all training details of their original implementation. For all methods, we employ a linear warm up and the Adam optimizer (Kingma and Ba, 2017), tuning the learning rates between $3e$-4 and $1e$-4, and weight decays between $1e$-2 and 0. All experiments are run on a NVIDIA A100 GPU.

It is worthy to mention that MuSEE can also build upon the decoder-only architecture by managing the KV cache and modifications to the position encodings (Xiao et al., 2024), though this requires additional management and is not the main focus of this study.

6 Results

In this section, we show the results for both quantitative and human side-by-side evaluation.

6.1 Quantitative Evaluation

Effectiveness comparison. The overall effectiveness comparison is shown in Table 1. We report traditional metrics, including precision, recall, and F1 score, in addition to our proposed AESOP metric. From the results, the MuSEE model consistently outperforms other baselines in terms of AESOP across all datasets. For instance, MuSEE achieves the highest AESOP scores on REBEL with 55.24 (T5-B) and 57.39 (T5-L), on NYT with 81.33 (T5-B) and 82.67 (T5-L), on CoNLL04 with 78.38 (T5-B) and 79.87 (T5-L), and on the Wikidata-based dataset with 46.95 (T5-B) and 50.94 (T5-L). These scores significantly surpass those of the competing models, indicating MuSEE's stronger entity extrac-

Table 1: Summary of results of different models. Each metric is shown in percentage (%). The last column shows the inference efficiency, measured by the number of samples the model can process per second. The best is **bolded**, and the second best is underlined. Our model has a statistical significance for $p \leq 0.01$ compared to the best baseline (labelled with *) based on the paired t-test.

Model	REBEL				NYT				CoNLL04				Wikidata-based				samples
	AESOP	Precision	Recall	F1	AESOP	Precision	Recall	F1	AESOP	Precision	Recall	F1	AESOP	Precision	Recall	F1	per sec
LM-JSON (T5-B)	41.91	38.33	**51.29**	43.87	66.33	73.10	52.66	61.22	68.80	61.63	48.04	53.99	36.98	43.95	29.82	35.53	19.08
GEN2OIE (T5-B)	44.52	35.23	40.28	37.56	67.04	72.08	53.02	61.14	68.39	62.35	42.20	50.26	37.07	40.87	28.37	33.55	28.21
IMoJIE (T5-B)	46.11	34.10	48.61	40.08	63.86	72.28	48.99	58.40	63.68	52.00	42.62	46.85	37.08	41.61	28.23	33.64	5.36
GenIE (T5-B)	48.82*	**57.55**	38.70	46.28*	79.41*	87.68	**73.24**	**79.81**	74.74*	72.49*	59.39	65.29	40.60*	50.27*	**29.75**	37.38	10.19
MuSEE (T5-B)	**55.24**	56.93	42.31	**48.54**	**81.33**	**88.29**	72.21	79.44	**78.38**	**73.18**	**60.28**	**66.01**	**46.95**	**53.27**	29.33	**37.99**	**52.93**
LM-JSON (T5-L)	45.92	39.49	40.82	40.14	67.73	73.38	53.22	61.69	68.88	61.50	47.77	53.77	38.19	43.24	31.63	36.54	11.24
GEN2OIE (T5-L)	46.70	37.28	41.12	39.09	68.27	73.97	53.32	61.88	68.52	62.76	43.31	51.16	38.25	41.23	28.54	33.77	18.56
IMoJIE (T5-L)	48.13	38.55	**49.73**	43.43	65.72	73.46	50.03	59.52	67.31	53.00	43.44	47.75	38.18	41.74	30.10	34.98	3.73
GenIE (T5-L)	50.06*	**58.00**	42.56	**49.09**	79.64*	84.82*	**75.69**	80.00	72.92*	**77.75**	55.64*	64.86	43.50*	**54.05**	30.98	**39.38**	5.09
MuSEE (T5-L)	**57.39**	57.11	42.89	48.96	**82.67**	**89.43**	73.32	**80.60**	**79.87**	74.89	**60.72**	**67.08**	**50.94**	53.72	**31.12**	39.24	**33.96**

tion capability. The other three traditional metrics further underscore the efficacy of the MuSEE model. For instance, on CoNLL04, MuSEE (T5-B) achieves a precision of 73.18, a recall of 60.28, and a F1 score of 66.01, which surpass all the other baselines. Similar improvements are observed on REBEL, NYT, and Wikidata-based dataset. Nevertheless, while MuSEE consistently excels in the AESOP metric, it does not invariably surpass the baselines across all the traditional metrics of precision, recall, and F1 score. Specifically, within the REBEL dataset, GenIE (T5-B) achieves the highest precision at 57.55, and LM-JSON (T5-B) records the best recall at 51.29. Furthermore, on the NYT dataset, GenIE (T5-B) outperforms other models in F1 score. These variances highlight the unique insights provided by our adaptive AESOP metric, which benefits from our entity-centric formulation. We expand on this discussion in section 6.2.

As discussed in Sec. 4, our MuSEE model is centered around two main enhancements: reducing output tokens and multi-stage parallel generation. By simplifying output sequences, MuSEE tackles the challenge of managing long sequences that often hinder baseline models, like LM-JSON, GenIE, IMoJIE, thus reducing errors associated with sequence length. Additionally, by breaking down the extraction process into three focused stages, MuSEE efficiently processes each aspect of entity extraction, leveraging contextual clues for more accurate predictions. In contrast, GEN2OIE's two-stage approach, though similar, falls short because it extracts relations first and then attempts to pair entities with these relations. However, a single relation may exist among different pairs of entities, which can lead to low performance with this approach. Supplemental ablation study is provided in Appendix E.

Efficiency comparison. As shown in the last column of Table 1, we provide a comparison on the inference efficiency, measured in the number of samples the model can process per second. The MuSEE model outperforms all baseline models in terms of efficiency, processing 52.93 samples per second with T5-B and 33.96 samples per second with T5-L. It shows a 10x speed up compared to IMoJIE, and a 5x speed up compared to the strongest baseline GenIE. This high efficiency can be attributed to MuSEE's architecture, specifically its multi-stage parallel generation feature. By breaking down the task into parallelizable stages, MuSEE minimizes computational overhead, allowing for faster processing of each sample. The benefit of this design can also be approved by the observation that the other multi-stage model, GEN2OIE, shows the second highest efficiency.

To better illustrate our model's strength, we show the scatter plots comparing all models with various backbones in Fig. 3 on the effectiveness and efficiency. We choose the Wikidata-based dataset and the effectiveness is measured by AESOP. As depicted, our model outperforms all baselines with a large margin. This advantage makes MuSEE particularly suitable for applications requiring rapid processing of large volumes of data, such as processing web-scale datasets, or integrating into interactive systems where response time is critical.

Grounding check. As the family of T5 models are pre-trained on Wikipedia corpus (Raffel et al., 2023), we are curious whether the models are extracting information from the given texts, or they are leveraging their prior knowledge to generate information that cannot be grounded to the given description. We use T5-L as the backbone in this experiment. We develop a simple approach to conduct this grounding check by perturbing the original test dataset with the following strategy. We first systematically extract and categorize all entities and their respective properties, based on their entity types. Then, we generate a perturbed version

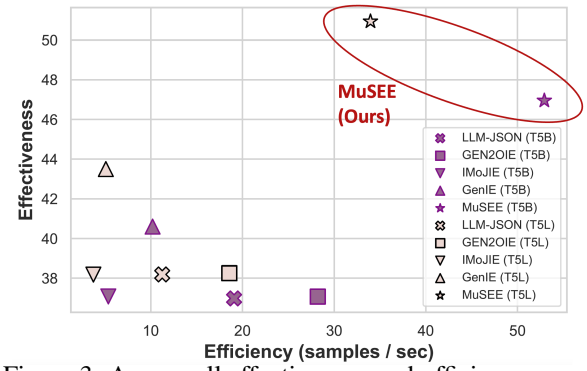

Figure 3: An overall effectiveness-and-efficiency comparison across models on Wikidata-based Dataset. MuSEE strongly outperforms all baselines on both measures. The effectiveness is measured by AESOP.

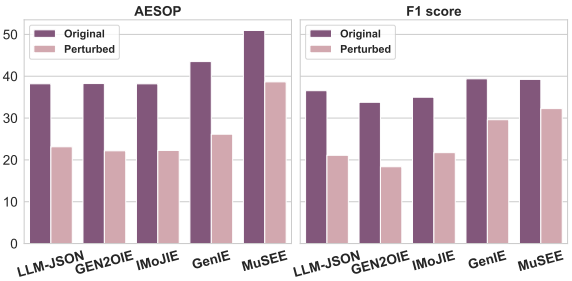

Figure 4: Grounding check across models on the Wikidata-based dataset. MuSEE shows the least performance drop on the perturbed version of data compared to other baselines.

of the dataset, by randomly modifying entity properties based on the categorization we built. We introduce controlled perturbations into the dataset by selecting alternative property values from the same category but different entities, and subsequently replacing the original values in the texts. The experiment results from our grounding study on the Wikidata-based dataset, as illustrated in Fig. 4, reveal findings regarding the performance of various models under the AESOP and F1 score. Our model, MuSEE, shows the smallest performance gap between the perturbed data and the original data compared to its counterparts, suggesting its stronger capability to understand and extract structured information from given texts.

6.2 Human Evaluation

To further analyze our approach, we randomly select 400 test passages from the Wikidata-based dataset, and generate outputs of our model MuSEE and the strongest baseline GenIE. Human evaluators are presented with a passage and two randomly flipped extracted sets of entities with properties. Evaluators are then prompted to choose the output they prefer or express no preference based on three criteria, *Completeness*, *Correctness*, and *Hallucinations* (details shown in Appendix F). Among all

	Human Evaluation			Quantitative Metrics			
	Complete.	Correct.	Halluc.	AESOP	Precision	Recall	F1
MuSEE prefer	61.75	59.32	57.13	61.28	45.33	37.24	40.57

Table 2: Percentage of samples preferred by humans and metrics on MuSEE's results when compared with GenIE's. The first three columns are for human evaluation. The next four columns are for quantitative metrics.

400 passages, the output of MuSEE is preferred 61.75% on the completeness, 59.32% on the correctness, and 57.13% on the hallucinations. For a complete comparison, we also report the percentage of samples preferred by quantitative metrics on MuSEE's results when compared with GenIE's, as summarized in Table 2. As shown, our proposed AESOP metric aligns more closely with human judgment than traditional metrics. These observations provide additional confirm to the quantitative results evaluated using the AESOP metric that our model significantly outperforms existing baselines and illustrates the inadequacy of traditional metrics due to their oversimplified assessment of extraction quality. Case study of the human evaluation is shown in Appendix F.

7 Discussion and Conclusion

We introduce Structured Entity Extraction (SEE), an entity-centric formulation of information extraction in a strict setting. We then propose the Approximate Entity Set OverlaP (AESOP) Metric, which focuses on the entity-level and more flexible to include different level of normalization. Based upon, we propose a novel model architecture, MuSEE, that enhances both effectiveness and efficiency. Both quantitative evaluation and human side-by-side evaluation confirm that our model outperforms baselines.

An additional advantage of our formulation is its potential to address coreference resolution challenges, particularly in scenarios where multiple entities share the same name or lack primary identifiers such as names. Models trained with prior triplet-centric formulation cannot solve the above challenges. However, due to a scarcity of relevant data, we were unable to assess this aspect in our current study.

8 Limitations

The limitation of our work lies in the assumption that each property possesses a single value. However, there are instances where a property's value might consist of a set, such as varying "names". Adapting our method to accommodate these scenarios presents a promising research direction.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Claire Cardie. 1997. Empirical methods in information extraction. *AI magazine*, 18(4):65–65.

Chia-Hui Chang, Mohammed Kayed, Moheb R Girgis, and Khaled F Shaalan. 2006. A survey of web information extraction systems. *IEEE transactions on knowledge and data engineering*, 18(10):1411–1428.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction.

Line Eikvil. 1999. Information extraction from world wide web-a survey. Technical report, Technical Report 945, Norweigan Computing Center.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ralph Grishman. 2015. Information extraction. *IEEE Intelligent Systems*, 30(5):8–15.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. Genie: Generative information extraction.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020a. Openie6: Iterative grid labeling and coordination analysis for open information extraction.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020b. Imojie: Iterative memory-based joint open information extraction.

Keshav Kolluru, Muqeeth Mohammed, Shubham Mittal, Soumen Chakrabarti, and Mausam. 2022. Alignment-augmented consistent translation for multilingual open information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2502–2517, Dublin, Ireland. Association for Computational Linguistics.

Shantanu Kumar. 2017. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Ruicheng Liu, Rui Mao, Anh Tuan Luu, and Erik Cambria. 2023. A brief survey on recent advances in coreference resolution. *Artificial Intelligence Review*, pages 1–43.

Keming Lu, Xiaoman Pan, Kaiqiang Song, Hongming Zhang, Dong Yu, and Jianshu Chen. 2023. Pivoine: Instruction tuning for open-world information extraction. *arXiv preprint arXiv:2305.14898*.

Jose L Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. 2020. Information extraction meets the semantic web: a survey. *Semantic Web*, 11(2):255–335.

Allan H Murphy. 1996. The finley affair: A signal event in the history of forecast verification. *Weather and forecasting*, 11(1):3–20.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2018. Information extraction from scientific articles: a survey. *Scientometrics*, 117:1931–1990.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878.

Italo L Oliveira, Renato Fileto, René Speck, Luís PF Garcia, Diego Moussallem, and Jens Lehmann. 2021. Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference*

on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Sunita Sarawagi et al. 2008. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377.

Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2022. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570.

Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2021. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Nikolaos Stylianou and Ioannis Vlahavas. 2021. A neural entity coreference resolution review. *Expert Systems with Applications*, 168:114466.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

Michael Vasilkovsky, Anton Alekseev, Valentin Malykh, Ilya Shenbin, Elena Tutubalina, Dmitriy Salikhov, Mikhail Stepnov, Andrey Chertok, and Sergey Nikolenko. 2022. Detie: Multilingual open information extraction inspired by object detection.

Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. 2018. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49.

Gerhard Weikum and Martin Theobald. 2010. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 65–76.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks.

Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, and Zhiwei Wang. 2022. A survey of information extraction based on deep learning. *Applied Sciences*, 12(19):9691.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. TextRunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA. Association for Computational Linguistics.

Hongbin Ye, Ningyu Zhang, Hui Chen, and Huajun Chen. 2022. Generative knowledge graph construction: A review. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1–17.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *arXiv preprint arXiv:2302.05019*.

A Variants of AESOP

The AESOP metric detailed in section 3.2 matches entities by considering all properties and normalizes with the maximum of the sizes of the target set and the predicted set. We denote it as AESOP-MultiProp-Max. In this section, we elaborate more variants of the AESOP metric in addition to section 3.2, categorized based on two criteria: the definition of entity similarity used for entity assignment and the normalization approach when computing the final metric value between $\mathcal{E}'$ and $\mathcal{E}$. These variants allow for flexibility and adaptability to different scenarios and requirements in structured entity extraction.

Variants Based on Entity Assignment. The first category of variants is based on the criteria for matching entities between the prediction $\mathcal{E}'$ and the ground-truth $\mathcal{E}$. We define three variants:

- **AESOP-ExactName**: Two entities are considered a match if their names are identical, disregarding case sensitivity. This variant is defined as $\mathbf{S}_{i,j} = 1$ if $v_{e'_i,\text{name}} = v_{e_j,\text{name}}$, otherwise 0.
- **AESOP-ApproxName**: Entities are matched based on the similarity of their "*name*" property values. This similarity can be measured using a text similarity metric, such as the Jaccard index.
- **AESOP-MultiProp**: Entities are matched based on the similarity of all their properties, with a much higher weight given to the "*entity name*" property due to its higher importance.

Variants Based on Normalization. The second category of variants involves different normalization approaches for computing the final metric value through Eq. 1:

- **AESOP-Precision**: The denominator is the size of the predicted set $\mathcal{E}'$, i.e., $\mu = m$.
- **AESOP-Recall**: The denominator is the size of the target set $\mathcal{E}$, i.e., $\mu = n$.
- **AESOP-Max**: The denominator is the maximum of the sizes of the target set and the predicted set, i.e., $\mu = \max\{m, n\}$.

Given these choices, we can obtain $3 \times 3 = 9$ variants of the AESOP metric. To avoid excessive complexity, we regard the AESOP-MultiProp-Max as default. For clarity, we illustrate the two phases of computing the AESOP metric and its variants in Fig. 5. We also show that precision and recall are specific instances of the AESOP metric in Appendix B.

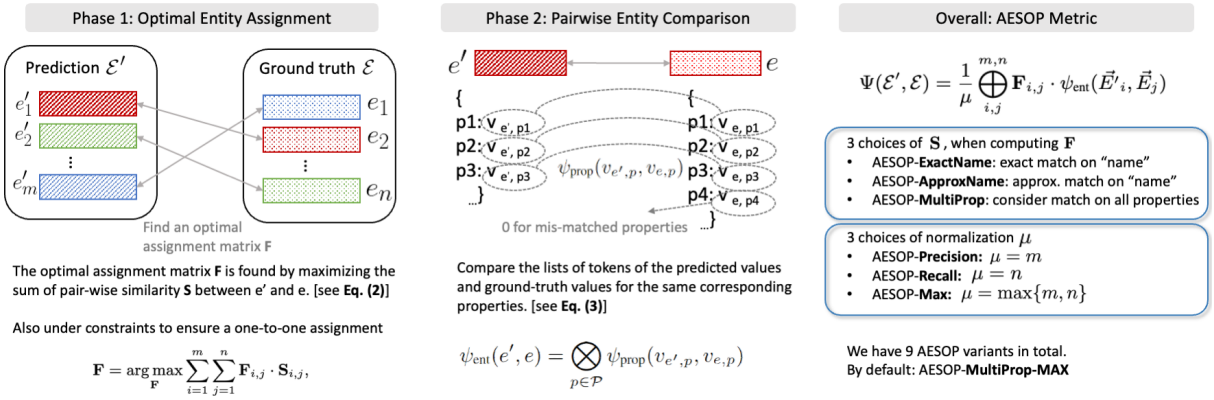

Figure 5: An illustration of the AESOP metric, including optimal entity assignment (phase 1) and pairwise entity comparison (phase 2), and overall metric computation with various similarity and normalization choices.

B Relationship between Precision/Recall and AESOP

In this section, we show the traditional metrics, precision and recall, are specific instances of the AESOP metric. To calculate precision and recall, we use the following equations on the number of triplets, where each triplet contains *subject*, *relation*, and *object*.

$$\text{precision} = \frac{\text{\# of correctly predicted triplets}}{\text{\# of triplets in the prediction}}, \tag{4}$$

$$\text{recall} = \frac{\text{\# of correctly predicted triplets}}{\text{\# of triplets in the target}}. \tag{5}$$

In the framework of the AESOP metric, precision and recall are effectively equivalent to treating each triplet as an entity, where the *subject* as the entity name, and the *relation* and *object* form a pair of property key and value. For optimal entity assignment (phase 1), precision and recall use the AESOP-MultiProp variant but match entities based on the similarity of all their properties with a same weight. For pairwise entity comparison (phase 2), the $\psi_{\text{ent}}(e', e)$ (Eq. 3), can be defined as 1 if $v' = v$, otherwise 0, where v' and v are the only property values in e' and e, respectively. For Eq. 1, $\bigoplus$ aggregation can be defined as a linear sum, which principally results in how many triplets are correctly predicted in this case. If μ in Eq. 1 is set as the number of triplets in the prediction, this corresponds to the calculation of precision. Similarly, when μ equals the number of triplets in the target, it corresponds to the calculation of recall.

C Details of Wikidata-based Dataset

We build a new Wikidata-based dataset. This dataset is inspired by methodologies employed in previous works such as Wiki-NRE (Trisedya et al., 2019), T-REx (Elsahar et al., 2018), REBEL (Huguet Cabot and Navigli, 2021), leveraging extensive information available on Wikipedia and Wikidata. The primary objective centers around establishing systematic alignments between textual content in Wikipedia articles, hyperlinks embedded within these articles, and their associated entities and properties as cataloged in Wikidata. This procedure is divided into three steps: *(i) Parsing Articles:* We commence by parsing English Wikipedia articles from the dump file[1], focusing specifically on text descriptions and omitting disambiguation and redirect pages. The text from each selected article is purified of Wiki markup to extract plain text, and hyperlinks within these articles are identified as associated entities. Subsequently, the text descriptions are truncated to the initial ten sentences, with entity selection confined to those referenced within this truncated text. This approach ensures a more concentrated and manageable dataset. *(ii) Mapping Wikidata IDs to English Labels:* Concurrently, we process the Wikidata dump[1] file to establish a mapping (termed as the *id-label map*) between Wikidata IDs and their corresponding English labels. This mapping allows for efficient translation of Wikidata IDs to their English equivalents. *(iii) Interconnecting Wikipedia articles with Wikidata properties:* For each associated entity within the text descriptions, we utilize Wikidatas API to ascertain its properties and retrieve their respective Wikidata IDs. The previously established *id-label map* is then employed to convert these property IDs into English labels. Each entitys type is determined using the value associated with *instance of (P31)*. Given the highly specific nature of these entity types (e.g., *small city (Q18466176)*, *town (Q3957)*, *big city (Q1549591)*), we implement a recursive merging process to generalize these types into broader categories, referencing the *subclass of (P279)* property. Specifically, we first construct a hierarchical taxonomy graph. Each node within this graph structure represents an entity type, annotated with a count reflecting the total number of entities it encompasses. Second, a priority queue are utilized, where nodes are sorted in descending order based on their entity count. We determine whether the top n nodes represent an ideal set of entity types, ensuring the resulted entity types are not extremely specific. Two key metrics are considered for this evaluation: the percentage of total entities encompassed by the top n nodes, and the skewness of the distribution of each entity type's counts within the top n nodes. If the distribution is skew, we then execute a procedure of dequeuing the top node and enqueueing its child nodes back into the priority queue. This iterative process allows for a dynamic exploration of the taxonomy, ensuring that the most representative nodes are always at the forefront. Finally, our Wikidata-based dataset is refined to contain the top-10 (i.e., $n = 10$) most prevalent entity types according to our hierarchical taxonomy graph and top-10 property keys in terms of occurrence frequency, excluding entity name and type. The 10 entity types are *talk, system, spatio-temporal entity, product, natural object, human, geographical feature, corporate body, concrete object,* and *artificial object*. The 10 property keys are *capital, family name, place of death, part of, location, country, given name, languages spoken, written or signed, occupation,* and *named after*.

[1]The version of the Wikipedia and Wikidata dump files utilized in our study are 20230720, representing the most recent version available during the development of our work.

D Statistics of Datasets

NYT is under the CC-BY-SA license. CoNLL04 is under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License. REBEL is under the Creative Commons Attribution 4.0 International License. The dataset statistics presented in Table 3 compare NYT, CoNLL04, REBEL, and Wikidata-based datasets. All datasets feature a minimum of one entity per sample, but they differ in their average and maximum number of entities, with the Wikidata-based dataset showing a higher mean of 3.84 entities. They also differ in the maximum number of entities, where REBEL has a max of 65. Property counts also vary, with REBEL having a slightly higher average number of properties per entity at 3.40.

Table 3: Statistics of all three datasets used in our paper.

Statistics	NYT	CoNLL04	REBEL	Wikidata-based
# of Entity Min	1	1	1	1
# of Entity Mean	1.25	1.22	2.37	3.84
# of Entity Max	12	5	65	20
# of Property Min	3	3	2	2
# of Property Mean	3.19	3.02	3.40	2.80
# of Property Max	6	4	17	8
# of Training Samples	56,196	922	2,000,000	23,477
# of Testing Samples	5,000	288	5,000	4,947

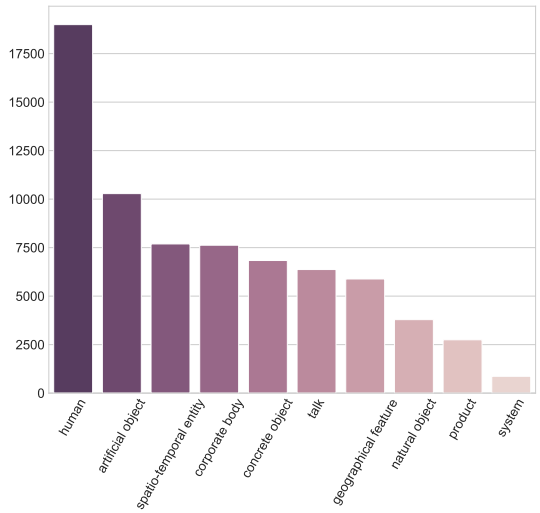

Figure 6: Frequency histogram of entity types in Wikidata-based Dataset.

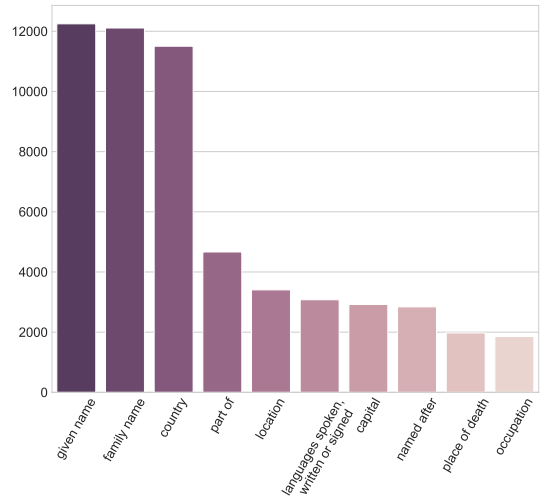

Figure 7: Frequency histogram of property keys in Wikidata-based Dataset.

E Ablation Study

The ablation study conducted on the MuSEE model, with the Wikidata-based dataset, serves as an evaluation of the model's core components: the introduction of special tokens and the Multi-stage parallel generation. By comparing the performance of the full MuSEE model against its ablated version, where only the special tokens feature is retained, we aim to dissect the individual contributions of these design choices to the model's overall efficacy. The ablated version simplifies the output format by eliminating punctuation such as commas, double quotes, and curly brackets, and by converting all entity types and property keys into special tokens. This is similar to the reducing output tokens discussed in Sec. 4. Results from the ablation study, as shown in Table 4, reveal significant performance disparities between the complete MuSEE model and its ablated counterpart, particularly when examining metrics across different model sizes (T5-B and T5-L) and evaluation metrics. The full MuSEE model markedly outperforms the ablated version across all metrics with notable improvements, underscoring the Multi-stage parallel

Table 4: Ablation study on Wikidata-based dataset. Each metric is shown in percentage (%).

Model	AESOP-ExactName			AESOP-ApproxName			AESOP-MultiProp		
	Max	Precision	Recall	Max	Precision	Recall	Max	Precision	Recall
w/o Multi-stage (T5-B)	25.19	40.87	27.64	25.75	42.14	28.26	26.93	44.49	29.72
MuSEE (T5-B)	**44.95**	**50.63**	**58.99**	**45.75**	**51.57**	**60.10**	**46.95**	**53.00**	**61.75**
w/o Multi-stage (T5-L)	27.74	53.04	28.81	28.14	54.10	29.22	29.14	56.90	30.29
MuSEE (T5-L)	**49.35**	**57.97**	**59.63**	**49.89**	**58.69**	**60.35**	**50.94**	**60.11**	**61.68**

generation's critical role in enhancing the model's ability to accurately and comprehensively extract entity-related information. These findings highlight the synergistic effect of the MuSEE model's design elements, demonstrating that both the Reducing output tokens and the Multi-stage parallel generation are pivotal for achieving optimal performance in structured entity extraction tasks.

F Human Evaluation Criteria and Case Study

The details for the three human evaluation criteria are shown below:

- *Completeness*: Which set of entities includes all relevant entities and has the fewest missing important entities? Which set of entities is more useful for further analysis or processing? Focus on the set that contains less unimportant and/or irrelevant entities.
- *Correctness*: Which set of entities more correctly represents the information in the passage? Focus on consistency with the context of the passage. Do extracted properties correctly represent each entity or are there more specific property values available? Are property values useful?
- *Hallucinations*: Which set of entities contains less hallucinations? That is, are there any entities or property values that do not exist or cannot be inferred from the text?

We provide a case study for the human evaluation analysis comparing the outputs of GenIE (T5-L) and MuSEE (T5-L) given a specific text description. MuSEE accurately identifies seven entities, surpassing GenIE's two, thus demonstrating greater completeness. Additionally, we identify an error in GenIE's output where it incorrectly assigns *Bartolomeo Rastrelli*'s place of death as *Moscow*, in contrast to the actual location, *Saint Petersburg*, which is not referenced in the text. This error by GenIE could stem from hallucination, an issue not present in MuSEE's output. In this example, it is evident that MuSEE outperforms GenIE in terms of *completeness*, *correctness*, and resistance to *hallucinations*.

Text Description: The ceremonial attire of Elizabeth, Catherine Palace, Tsarskoye Selo; fot. Ivonna Nowicka Elizabeth or Elizaveta Petrovna (;) reigned as Empress of Russia from 1741 until her death in 1762. She remains one of the most popular Russian monarchs because of her decision not to execute a single person during her reign, her numerous construction projects, and her strong opposition to Prussian policies. The second-eldest daughter of Tsar Peter the Great (), Elizabeth lived through the confused successions of her father's descendants following her half-brother Alexei's death in 1718. The throne first passed to her mother Catherine I of Russia (), then to her nephew Peter II, who died in 1730 and was succeeded by Elizabeth's first cousin Anna (). After the brief rule of Anna's infant great-nephew, Ivan VI, Elizabeth seized the throne with the military's support and declared her own nephew, the future Peter III, her heir. During her reign Elizabeth continued the policies of her father and brought about a remarkable Age of Enlightenment in Russia. Her domestic policies allowed the nobles to gain dominance in local government while shortening their terms of service to the state. She encouraged Mikhail Lomonosov's foundation of the University of Moscow, the highest-ranking Russian educational institution. Her court became one of the most splendid in all Europe, especially regarding architecture: she modernised Russia's roads, encouraged Ivan Shuvalov's foundation of the Imperial Academy of Arts, and financed grandiose Baroque projects of her favourite architect, Bartolomeo Rastrelli, particularly in Peterhof Palace.

GenIE (T5-L):

```
{
    "0": {
        "entity name": "Bartolomeo Rastrelli",
        "place of death": "Moscow",
        "given name": "Bartolomeo",
        "family name": "Rastrelli",
        "languages spoken, written or signed": "Russian",
        "country": "Russia",
        "type": "human"
    },
    "1": {
        "entity name": "Catherine I of Russia",
        "given name": "Catherine",
        "languages spoken, written or signed": "Russian",
        "type": "human"
    }
}
```


MuSEE (T5-L):


```
{
    "0": {
        "type": "human",
        "entity name": "Peter the Great",
        "given name": "Peter"
    },
    "1": {
        "type": "human",
        "entity name": "Bartolomeo Rastrelli",
        "family name": "Rastrelli",
        "given name": "Bartolomeo"
    },
    "2": {
        "type": "human",
        "entity name": "Catherine I of Russia",
        "given name": "Catherine"
    },
    "3": {
        "type": "human",
        "entity name": "Peter III",
        "given name": "Peter"
    },
    "4": {
        "type": "human",
        "entity name": "Ivan Shuvalov",
        "family name": "Shuvalov",
        "given name": "Ivan",
        "languages spoken, written or signed": "Russian"
    },
    "5": {
        "type": "concrete object",
        "entity name": "Peterhof Palace",
```

```
        "country": "Russia"
    },
    "6": {
        "type": "human",
        "entity name": "Mikhail Lomonosov",
        "family name": "Lomonosov",
        "given name": "Mikhail",
        "languages spoken, written or signed": "Russian"
    }
}
```


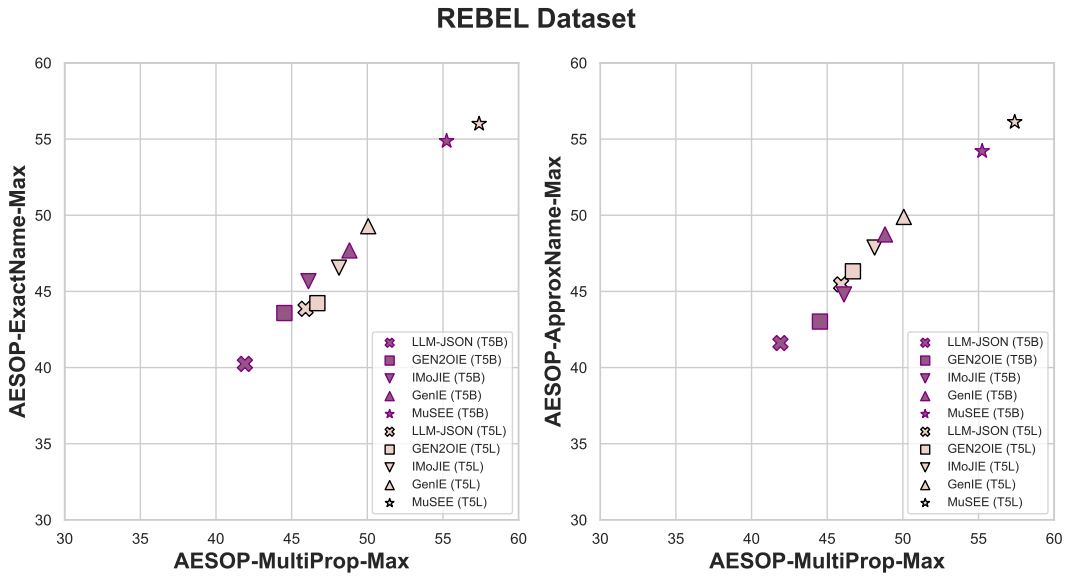

Figure 8: Metric correlation analysis on the REBEL dataset.

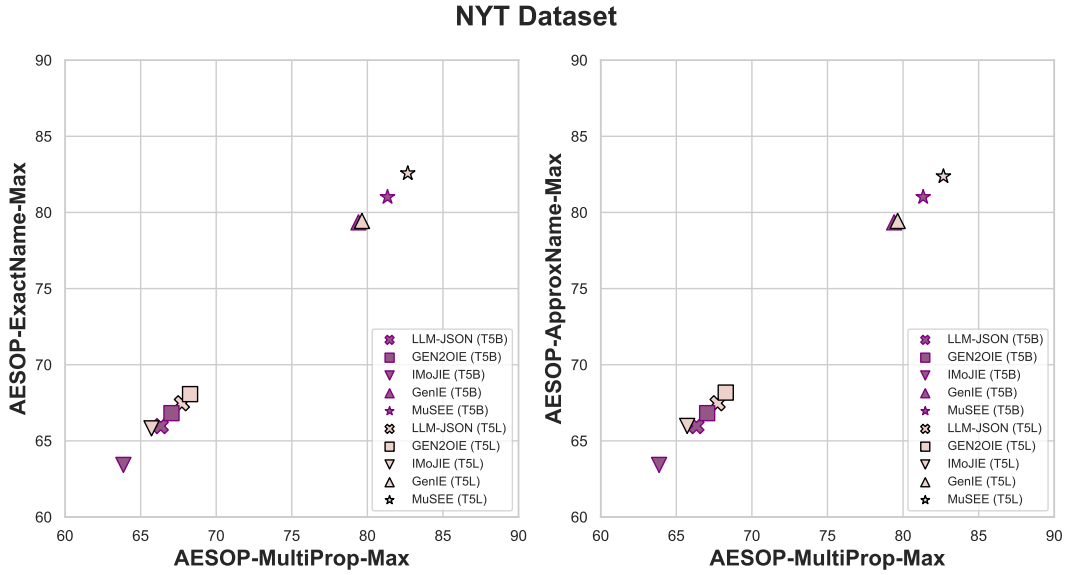

Figure 9: Metric correlation analysis on the NYT dataset.

G Metric Correlation Analysis

We show the correlation analysis between AESOP metric variants across all models on all four datasets, shown in Fig. 8, Fig. 9, Fig. 10, and Fig. 11, respectively. Specifically, we focus on the correlation analysis

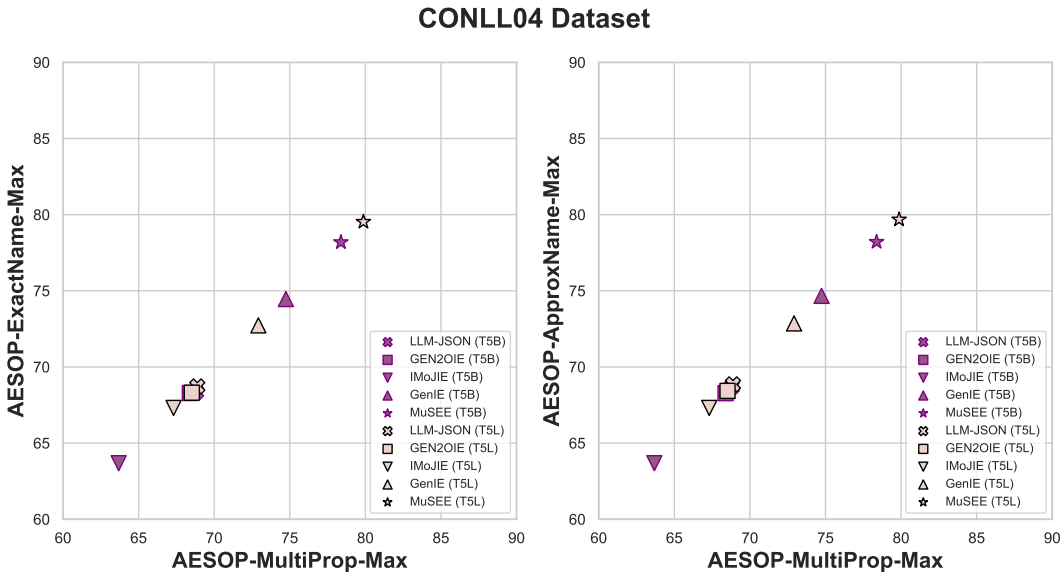

Figure 10: Metric correlation analysis on the CONLL04.

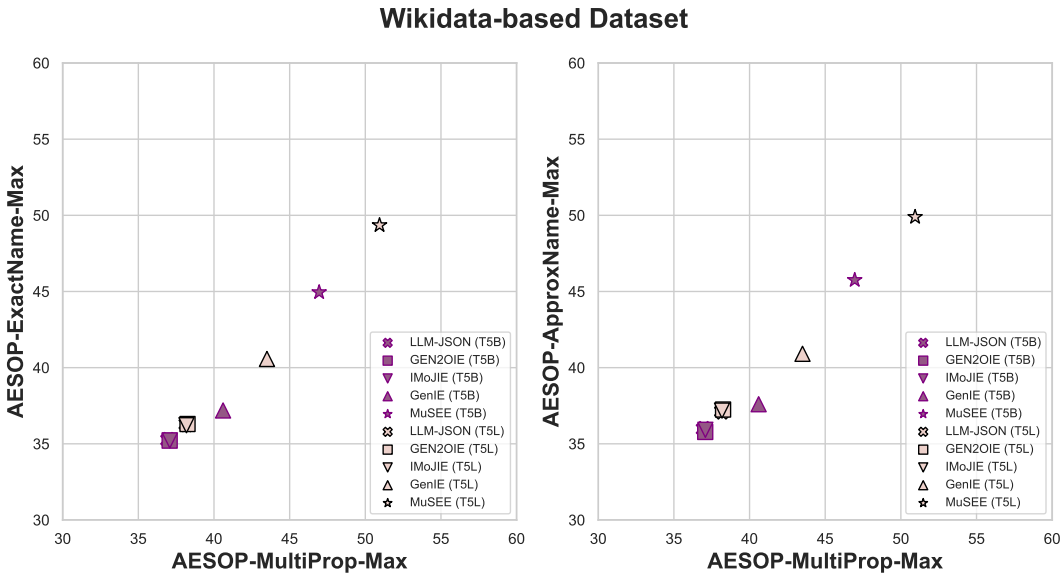

Figure 11: Metric correlation analysis on the Wikidata-based dataset.

of different variants based on entity assignment variants in Phase 1 of AESOP, as described in Sec. 3. For Phase 2, the "Max" normalization method is employed by default. Observations for the other two normalization variants are similar. In the associated figures, AESOP-MultiProp-Max is uniformly used as the x-axis measure, while AESOP-ExactName-Max or AESOP-ApproxName-Max serve as the y-axis metrics. The scatter plots in all figures tend to cluster near the diagonal, indicating a robust correlation among the various metric variants we have introduced.